



#4

## PATENTS

## UNITED STATES PATENT AND TRADEMARK OFFICE

S/N: 09/943,786

Filed: 08/31/01

Inventor: Simpson et. al

Title: Method and Apparatus for  
Presenting, Searching, and Viewing  
Directories

Art Unit: 2151

Docket No.: NO078-100002

## PRELIMINARY AMENDMENT

Assistant Commissioner for Patents  
Washington, D.C. 20231

Dear Sir:

AMENDMENT

## IN THE CLAIMS

Please rewrite the following claims as follows:

1. (Amended) A computer system, comprising:
  - one or more directories comprising a plurality of directory objects each having one or more object attributes and data;
  - a directory shell referencing said one or more directories;
  - one or more categories in said directory shell, each of said categories being associated with one or more directory objects and at least a portion of the associated object attributes;
  - a query interface operable to query the categories in the directory shell; and
  - a directory interface operable to send a search request to the one or more directories in accordance with the query, and receive data from the one or more directories that satisfies the search request.

91 701100

100.00 CL

2. (Amended) The computer system of claim 1, wherein at least one of the directories is LDAP compliant.
5. (Amended) The computer system of claim 1, wherein the query interface is formatted in HTML.
6. (Amended) The computer system of claim 1, further comprising an administrator utility to configure the categories and the query interface.
7. (Amended) A method in a computer system, comprising:
  - creating a directory shell comprising one or more categories;
  - associating said directory shell with one or more directories, wherein each directory comprises a plurality of directory objects having object attributes and attribute data;
  - associating each of said categories with one or more directory objects and at least a portion of the object attributes corresponding to the directory objects;
  - requesting a search for query data against a selected category; and
  - searching the one or more directories for the query data against the attribute data corresponding to the directory objects and object attributes associated with the selected category.
8. (Amended) The method of claim 7, wherein associating categories includes mapping category attributes with object attributes.
9. (Amended) The method of claim 7, wherein [the step of] requesting a search includes formatting a Boolean search.
10. (Amended) The method of claim 7, wherein the creating, associating with directories, associating with directory objects, requesting, and searching are performed sequentially.

11. (Amended) A computer readable medium comprising a plurality of computer instructions for performing a method comprising:
- creating a directory shell comprising one or more categories;
  - associating said directory shell with one or more directories, wherein each directory comprises a plurality of directory objects having object attributes and attribute data;
  - associating each of said categories with one or more directory objects and at least a portion of the object attributes corresponding to the directory objects;
  - requesting a search for query data against a selected category; and
  - searching the one or more directories for the query data against the attribute data corresponding to the directory objects and object attributes associated with the selected category.
12. (Amended) A propagated signal comprising a plurality of computer instructions for performing a method comprising:
- creating a directory shell comprising one or more categories;
  - associating said directory shell with one or more directories, wherein each directory comprises a plurality of directory objects having object attributes and attribute data;
  - associating each of said categories with one or more directory objects and at least a portion of the object attributes corresponding to the directory objects;
  - requesting a search for query data against a selected category; and
  - searching the one or more directories for the query data against the attribute data corresponding to the directory objects and object attributes associated with the selected category.
13. (Amended) A data structure, comprising:
- one or more directories, each directory comprising a plurality of classes with attributes and a plurality of objects instantiated from

said classes, wherein said objects comprise a plurality of data associated with the attributes;

- a directory shell associated with said one or more directories;
- a plurality of categories associated with said directory shell, wherein each category corresponds to one or more classes in the one or more directories;
- a plurality of category attributes associated with each category, wherein each category attribute corresponds to an attribute of the class corresponding to the associated category; and

wherein the directory shell is querable against the categories to search and retrieve data of the objects in the one or more directories.

17. (Amended) The data structure of claim 13, wherein the categories and category attributes are object oriented programming classes.
19. (Amended) The data structure of claim 13, wherein the directory shell includes one or more directory references each associated with at least one of the one or more directories.

## IN THE SPECIFICATION

Please delete and replace the following paragraphs as follows:

Page 2, paragraph starting at line 9:

The present invention relates generally to computer systems and software, and will be specifically disclosed as a method and apparatus for presenting, searching, and viewing directories.

Page 2, paragraph starting at line 14:

The virtual explosion of technical advances in microelectronics, digital computers and software have changed the face of modern society. In fact, these technical advances have become so important and pervasive that this explosion is sometimes referred to as "The Information Revolution." Through

telephone lines, networks, satellite communications and the like, information and resources are ever increasingly being accessed and shared.

Page 3, paragraph starting at line 13:

In many cases, the structure of a directory is governed by a schema. The schema defines the rules for adding and managing objects and object attributes in a directory. For instance, a schema can be specified through a data dictionary that provides a standard set of data types or classes from which objects can be created. In such a case, each object in the directory belongs to a class that specifies which attributes are associated with the object. In other words, an object is of a particular type and is created from a corresponding class. For example, a User object would be created from a User class, a Printer object would be created from a Printer class, etc. An object created from a class is sometimes referred to as being "instantiated" from the class, wherein the object is referred to as an "instance" of the class. Generally, such schemas are extensible so that they may be tailored to modify existing classes or add new classes.

Page 4, paragraph starting at line 3:

There are a variety of types, formats and standards for directories. For example, some directories are compliant with the Lightweight Directory Access Protocol ("LDAP"). LDAP is a protocol used to access a directory. One common use for LDAP compliant directories is in conjunction with Web browsers and e-mail programs, however, LDAP directories are used in a variety of other applications as well. Another example is X.500 directories, which are hierarchal directories that use the Directory Access Protocol ("DAP") to access such directories. Beyond these two limited examples, a variety of other types of directories are currently known to one with ordinary skill in the art. Moreover, additional types of directories will undoubtedly emerge in the future. Some current examples of directories include Novell Directory Services "NDS", Novell's eDirectory, Oracle Databases, Netscape's Directory Server and Planet Directory Server, Microsoft's Active Directory, IBM's SecureWay Directory.

Page 7, paragraph starting at line 23:

A directory shell application 40 runs on the computer 30, which, for example, may have been received by the computer 30 on a computer readable medium 32 or propagated signal 34. The directory shell 40 includes references to the directories 10 and includes one or more categories. Each category is associated with one or more objects in the directories 10. The categories may also be associated with all or a subset of the object attributes in the directories 10. A directory interface 46 formats, submits, and receives data to/from the directories 10. As different directories have different protocols, it is preferred that the directories 10 each be LDAP compliant, however, if a given directory 10 is not LDAP compliant, the directory drivers 42 can provide the necessary format/protocol information so that the directory interface 46 can communicate with such directory. A user interface 44 allows a user to view, search and interact with the directories 10. For instance, the user interface 44 could allow the user to format a query directed to objects in the directories 10 and present the object data that satisfy the query. Preferably, such queries are formatted against the categories. The user interface 44 could take a variety of forms known as the art, including an application device (e.g. cell phone, personal digital assistant "PDA", and the like), applet, servlet, function in a library, hypertext makeup language, extensible makeup language, wireless makeup language, dynamic hypertext makeup language, and the like. Further, the user interface 44 can operate on any device receiving wired or wireless data, including workstations, PDA's, cell phones, computers, and the like. The directory interface 46, directory shell 40, and user interface 44 can be an integrated set of instructions, a separate module of instructions, part of a library of instructions, any combination of the foregoing, or in other formats known in the art.

Page 8, paragraph starting at line 24:

The directory shell 60 comprises one or more directory references 61. Each directory reference 61 is associated with a directory 50 and can include a variety of data specific to the directory 50. The directory references 61 identify which directories the directory shell 60 will access. The directory shell 60 also includes a collection of categories 62. Preferably, each directory

reference 61 has its own unique set of associated categories 62. A category 62 represents and references one or more directory classes 51 that have been grouped together under one category name. In other words, any given category 62 is a collection of directory classes 51 and the directory objects 53 instantiated from such directory classes 51. A category 62 can reference classes 51 in a single directory or classes of different directories. For instance, a category called "Find People" can be configured to search only the Novell directory for objects instantiated from a User directory class. In another example, a category called "Cisco People" can be configured to search only the Cisco directory for User objects. In another example, a category called "Find All" can be configured to search both the Novell and Cisco directories for User objects. As one with ordinary skill in the art will recognize, the combinations and flexibility of categories 62 are virtually unlimited.

Page 9, paragraph starting at line 17:

Fig. 3 illustrates a method 70 utilizing the data structures 50 and 60. At step 71, a directory shell 60 is created. At step 72, one or more directories 50 are associated with the directory shell 60. For instance, a directory reference 61 can be created for each directory 50 associated with the directory shell 60. During step 73, a plurality of categories 62 are associated with directory objects 53. For instance, each category 62 could reference one or more directory classes 51 such that directory objects 53 instantiated from such directory classes 51 are associated to the category 62. At step 74, category attributes 63 are associated or mapped to object attributes, such as to the class attributes 52, which are included in an instantiated directory object 53. At step 75, a search request is sent to the directory 50 based on the categories 62 and category attributes 63 for the corresponding directory reference 61. For instance, a query can be formatted such that query data is matched against a selected category and the category attributes. In such an instance, the category may be referred to as a search object. The formatted query could then be translated to search for the query data against the directory objects in the directory that corresponds to the selected category. Function calls using LDAP may be employed to transmit the request to the

directory, assuming the directory is LDAP compliant. In another example, function calls could be prepared in accordance with appropriate directory drivers. The directory 50 replies to the search request and object data 54 satisfying the search is returned to the directory shell 60 during step 76. At step 77, the received object data 54 is presented, preferably in a user interface associated with the directory shell 60. It should be noted that the sequence of the steps in method 70 are presented as an illustration and the sequence may be varied.

Page 10, paragraph starting at line 7:

By way of example, the following illustrates a data structure created from Java classes, wherein an instantiated class becomes an object from that class. Each Java class comprises data members and method members.

Page 11, paragraph starting at line 12:

In one example, a directory shell, which operates in conjunction with the foregoing Java class structure, is used in conjunction with Novell's eGuide product. This example of a directory shell includes two aspects: an administration utility and a directory browser. The administration utility is generally used by system administrators to configure and manage the directory shell. The directory browser, is typically used by end users to search and view directories, and if appropriate rights are granted, modify the directory. In the present example, the user interface for both aspects of the directory shell are generated by servlets and are presented in hypertext markup language ("HTML") and viewed in a standard browser, however, a variety of alternative user interfaces and variations are also contemplated, such as stand alone applications, applets, wireless devices, and the like.

Page 12, paragraph starting at line 23:

One advantage of the Authentication Group functionality is that it utilizes a user's credentials for LDAP queries (inside the Authentication Group), thus utilizing the Access Control List ("ACL"). In regard to consuming the ACL, critical information can easily be protected and controlled through directory rights. Directories containing "general" or "non-critical" information



can simply be excluded from the Authentication Group, where proxy credentials will be sufficient for viewing its data. This provides many different powerful business-to-business solutions.

Page 15, paragraph starting at line 30:

The Photos Properties option 112 configures the use of photographs in directory shell. The user may specify where image files are stored, such as on a directory or on a file system, the maximum image size, whether a photo agreement is required, and whether users are allowed to upload images.

Page 16, paragraph starting at line 29:

The Restrictions option 112 configures whether validated credentials will be stored on the client machine in a cookie using Base64 encryption. Subsequent loading of the directory shell will automatically authenticate the client with the credentials stored in the cookie. An administrator can also specify how many seconds until the cookie (if enabled and exists) will expire. Further, an administrator can specify whether directory authentication is required prior to the directory browser being available.

Page 19, paragraph starting at line 6:

The foregoing description and examples of the invention have been presented for purposes of illustration and description. These are not intended to be exhaustive nor to limit the invention to the precise form disclosed. Many alternatives, modifications, and variations will be apparent to those skilled in the art in light of the above teaching. Accordingly, this invention is intended to embrace all alternatives, modifications, and variations that fall within the spirit and broad scope of the appended claims.

## IN THE ABSTRACT

Please delete and replace the Abstract as follows:

A directory shell references one or more directories. Each directory comprises a plurality of directory objects each having object attributes and

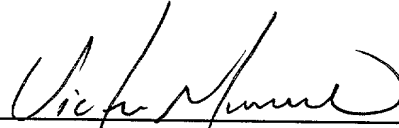
attribute data. One or more categories in the directory shell are associated with one or more directory objects and at least a portion of the associated object attribute. A user interface is operable for a user to query the categories in the directory shell. A directory interface is operable to send a search request to the directories in accordance with the query and receive data from the directories that satisfy the search request.

### REMARKS

Claims 1, 2, 5-13, 17 and 19 have been amended. The amendments are being made to correct typographical errors and for business reasons. The amendments are fully supported by the specification, claims, and figures as originally filed. No new matter is believed or intended to be involved. In accordance with 37 CFR 1.121, attached is mark-up version of the changes made in the present amendment captioned "Marked-Up Version of the Amendments".

Applicant has submitted with this paper substitute drawings in response to the Notice to File Corrected Application Papers mailed on October 5, 2001. The substitute drawings are identical to the drawings originally filed, but have corrected the margin informality specified in the Notice.

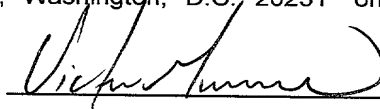
Respectfully submitted,



Victor Moreno (Reg. No. 40,732)  
Graydon Head & Ritchey LLP  
1900 Fifth Third Center  
511 Walnut Street  
Cincinnati, Ohio 45202-3157  
(513) 621-6464  
ATTORNEYS OF RECORD

**Certificate of Mailing**

The undersigned certifies that this correspondence was deposited with the U.S. Postal Service with sufficient postage as first class mail and addressed to "Assistant Commissioner for Patents, Washington, D.C., 20231" on Dec 5, 2001.



## MARK-UP VERSION OF THE AMENDMENTS

### IN THE CLAIMS

1. (Amended) A computer system, comprising:
  - one or more directories [accessible by said computer, said one or more directories] comprising a plurality of directory objects[,]  
each [object] having one or more object attributes and data;
  - a directory shell referencing said one or more directories;
  - one or more categories in said directory shell, each of said categories being associated with one or more directory objects and at least a portion of the associated object attributes;
  - a [user] query interface operable [for a user] to query the categories in the directory shell; and
  - a directory interface operable to send a search request to the one or more directories in accordance with the query, and receive data from the one or more directories that satisfies the search request.
2. (Amended) The computer system of claim 1, wherein at least one of the directories [are] is LDAP compliant.
5. (Amended) The computer system of claim 1, wherein the [user] query interface is formatted in HTML.
6. (Amended) The computer system of claim 1, further comprising an administrator utility to configure the categories and the [user] query interface.
7. (Amended) A method in a computer system, comprising [the steps of]:
  - creating a directory shell comprising one or more categories;
  - associating said directory shell with one or more directories, wherein each directory comprises a plurality of directory objects having object attributes and attribute data;

- associating each of said categories with one or more directory objects and at least a portion of the object attributes corresponding to the directory objects;
- requesting a search for query data against a selected category; and
- searching the one or more directories for the query data against the attribute data corresponding to the directory objects and object attributes associated with the selected category[;
- receiving attribute data satisfying the search; and
- presenting the received attribute data].

8. (Amended) The method of claim 7, wherein [the step of] associating categories [further comprise] includes mapping category attributes with object attributes.
9. (Amended) The method of claim 7, wherein [the step of] requesting a search [comprises] includes formatting a Boolean search.
10. (Amended) The method of claim 7, wherein the [steps] creating, associating with directories, associating with directory objects, requesting, and searching are performed sequentially [as listed].
11. (Amended) A computer readable medium comprising a plurality of computer instructions for performing [the] a method [of claim 7] comprising:
  - creating a directory shell comprising one or more categories;
  - associating said directory shell with one or more directories, wherein each directory comprises a plurality of directory objects having object attributes and attribute data;
  - associating each of said categories with one or more directory objects and at least a portion of the object attributes corresponding to the directory objects;

- requesting a search for query data against a selected category;  
and
- searching the one or more directories for the query data against  
the attribute data corresponding to the directory objects and  
object attributes associated with the selected category.

12. (Amended) A propagated signal comprising a plurality of computer instructions for performing [the] a method [of claim 7] comprising:

- creating a directory shell comprising one or more categories;
- associating said directory shell with one or more directories,  
wherein each directory comprises a plurality of directory objects  
having object attributes and attribute data;
- associating each of said categories with one or more directory  
objects and at least a portion of the object attributes  
corresponding to the directory objects;
- requesting a search for query data against a selected category;  
and
- searching the one or more directories for the query data against  
the attribute data corresponding to the directory objects and  
object attributes associated with the selected category.

13. (Amended) A data structure, comprising:

- one or more directories, each directory comprising a plurality of classes with attributes and a plurality of objects instantiated from said classes, wherein said objects comprise a plurality of data associated with the attributes;
- a directory shell associated with said one or more directories;
- a plurality of categories associated with said directory shell, wherein each category corresponds to one or more classes in the one or more directories;
- a plurality of category attributes associated with each category, wherein each category attribute corresponds to an attribute of the class corresponding to the associated category; and

wherein the directory shell is querable against the categories to search and retrieve data of the objects in the one or more directories.

17. (Amended) The data structure of claim 13, wherein the categories and category attributes are object[ed] oriented programming classes.
19. (Amended) The data structure of claim 13, wherein the directory shell [further comprises] includes one or more directory references each associated with at least one of the one or more directories.

## IN THE SPECIFICATION

Page 2, paragraph starting at line 9:

The present invention relates generally to computer systems and software, and will be specifically disclosed as a method and apparatus for presenting, searching, and viewing directories.

Page 2, paragraph starting at line 14:

The virtual explosion of technical advances in microelectronics, digital computers and software have changed the face of [modem] modern society. In fact, these [technological] technical advances have become so important and pervasive that this explosion is sometimes referred to as "[the information revolution] The Information Revolution." Through telephone lines, networks, satellite communications and the like, information and resources are ever increasingly being accessed and shared.

Page 3, paragraph starting at line 13:

In many cases, the structure of a directory is governed by a schema. The schema defines the rules for adding and managing objects and object attributes in a directory. For instance, a schema can be specified through a data dictionary that provides a standard set of data types or classes from which objects can be created. In such a case, each object in the directory belongs to a class that specifies which attributes are associated with the

object. In other words, an object is of a particular type and is created from a corresponding class. For example, a User object would be created from a User class, a Printer object would be created from a Printer class, etc. An object created from a class is sometimes referred to as being "instantiated" from the class, wherein the object is referred to as an "instance" of the class. Generally, such schemas are extensible so that [it] they may be tailored to modify existing classes or add new classes.

Page 4, paragraph starting at line 3:

There are a variety of types, formats and standards for directories. For example, some directories are compliant with the Lightweight Directory Access Protocol ("LDAP"). LDAP is a protocol used to access a directory. One common use for LDAP compliant directories is in conjunction with Web browsers and e-mail programs, however, LDAP directories are used in a variety of other applications as well. Another example is X.500 directories, which are hierarchal directories that use the Directory Access Protocol ("DAP") [protocol] to access such directories. Beyond these two limited examples, a variety of other types of directories are currently known to one with ordinary skill in the art. Moreover, additional types of directories will undoubtedly emerge in the future. Some current examples of directories include Novell Directory Services "NDS", Novell's eDirectory, Oracle Databases, Netscape's Directory Server and Planet Directory Server, Microsoft's Active Directory, IBM's SecureWay Directory.

Page 7, paragraph starting at line 23:

A directory shell application 40 runs on the computer 30, which, for example, may have been received by the computer [40] 30 on a computer readable medium 32 or propogated signal 34. The directory shell 40 includes references to the directories 10 and includes one or more categories. Each category is associated with one or more objects in the directories 10. The categories may also be associated with all or a subset of the object attributes in the directories 10. A directory interface 46 formats, submits, and receives data to/from the directories 10. As different directories have different protocols, it is preferred that the directories 10 each be LDAP compliant,



however, if a given directory 10 is not LDAP compliant, the directory drivers [34] 42 can provide the necessary format/protocol information so that the directory interface 46 can communicate with such directory. A user interface 44 allows a user to view, search and interact with the directories 10. For instance, the user interface 44 could allow the user to format a query directed to objects in the directories 10 and present the object data that satisfy the query. Preferably, such queries are formatted against the categories. The user interface 44 could take a variety of forms known as the art, including an application device (e.g. cell phone, personal digital assistant "PDA", and the like), applet, [serverlet] servlet, function in a library, hypertext makeup language, extensible makeup language, wireless makeup language, dynamic hypertext makeup language, and the like. Further, the user interface 44 can operate on any device receiving wired or wireless data, including workstations, PDA's, cell phones, computers, and the like. The directory interface 46, directory shell 40, and user interface 44 can be an integrated set of instructions, a separate module of instructions, part of a library of instructions, any combination of the foregoing, or in other formats known in the art.

Page 8, paragraph starting at line 24:

The directory shell 60 comprises one or more directory references 61. Each directory reference 61 is associated with a directory 50 and can include a variety of data specific to the directory 50. The directory references 61 identify which directories [that] the directory shell 60 will access. The directory shell 60 also includes a collection of categories 62. Preferably, each directory reference 61 has its own unique set of associated categories 62. A category 62 represents and references one or more directory classes 51 that have been grouped together under one category name. In other words, any given category 62 is a collection of directory classes 51 and the directory objects 53 instantiated from such directory classes 51. A category 62 can reference classes 51 in a single directory or classes of different directories. For instance, a category called "Find People" can be configured to search only the Novell directory for objects instantiated from a User directory class. In another example, a category called "Cisco People" can be configured to

search only the Cisco directory for User objects. In another example, a category called "Find All" can be configured to search both the Novell and Cisco directories for User objects. As one with ordinary skill in the art will recognize, the combinations and flexibility of categories 62 [is] are virtually unlimited.

Page 9, paragraph starting at line 17:

Fig. 3 illustrates a method 70 utilizing the data structures 50 and 60. At step 71, a directory shell 60 is created. At step 72, one or more directories 50 are associated with the directory shell 60. For instance, a directory reference 61 can be created for each directory 50 associated with the directory shell 60. During step 73, a plurality of categories 62 are associated with directory objects 53. For instance, each category 62 could reference one or more directory classes 51 such that directory objects 53 instantiated from such directory classes 51 are associated to the category 62. At step 74, category attributes 63 are associated or mapped to object attributes, such as to the class attributes 52, which are included in an instantiated directory object 53. At step 75, a search request [to] is sent to the directory 50 based on the categories 62 and category attributes 63 for the corresponding directory reference 61. For instance, a query can be formatted such that query data is matched against a selected category and the category attributes. In such an instance, the category may be referred to as a search object. The formatted query could then be translated to search for the query data against the directory objects in the directory that corresponds to the selected category. Function calls using LDAP may be employed to transmit the request to the directory, assuming the directory is LDAP compliant. In another example, function calls could be prepared in accordance with appropriate directory drivers. The directory 50 replies to the search request and object data 54 satisfying the search is returned to the directory shell 60 during step 76. At step 77, the received object data 54 is presented, [preferable] preferably in a user interface associated with the directory shell 60. It should be noted that the sequence of the steps in method 70 are presented as an illustration and the sequence may be varied.

Page 10, paragraph starting at line 7:

By way of example, the following illustrates a data structure created from Java classes, wherein an instantiated class becomes an object from that class[:]. Each Java class comprises data members and method members.

Page 11, paragraph starting at line 12:

[One] In one example, a directory shell, which operates in conjunction with the foregoing Java class structure, is used in conjunction with Novell's eGuide product. This example of a directory shell includes two aspects: an administration utility and a directory browser. The administration utility is generally used by system administrators to [configuring and managing] configure and manage the directory shell. The directory browser, is typically used by end users to search and view directories, and if appropriate rights are granted, modify the directory. In the present example, the user interface for both aspects of the directory shell are generated by [serverlets] servlets and are presented in hypertext markup language ("HTML") and viewed in a standard browser, however, a variety of alternative user interfaces and variations are also contemplated, such as stand alone applications, applets, wireless devices, and the like.

Page 12, paragraph starting at line 23:

One advantage of the Authentication Group functionality is that it utilizes a user's credentials for LDAP queries (inside the Authentication Group), thus utilizing the Access Control List ("ACL"). In regard to consuming the ACL, critical information can easily be protected and controlled through directory rights. Directories containing "general" or "non-critical" information can simply be excluded from the Authentication Group, where proxy credentials will be sufficient for viewing its data. This [poses] provides many different powerful business-to-business solutions.

Page 15, paragraph starting at line 30:

The Photos Properties option 112 configures the use of photographs in directory shell. The user may specify where image files are [stores] stored, such as on a directory or on a file system, the maximum image size, whether

a photo agreement is required, and whether users are allowed to upload images.

Page 16, paragraph starting at line 29:

The Restrictions option 112 configures whether validated credentials will be stored on the client machine in a cookie using Base64 encryption. Subsequent loading of the directory shell will automatically authenticate the client with the credentials stored in the cookie. An administrator can also specify how many seconds until the cookie (if enabled and exists) will expire. Further, an administrator can specify whether directory authentication is required prior to the directory browser being available.

Page 19, paragraph starting at line 6:

The foregoing description and examples of the invention [has] have been presented for purposes of illustration and description. [It is] These are not intended to be exhaustive nor to limit the invention to the precise form disclosed. Many alternatives, modifications, and variations will be apparent to those skilled in the art in light of the above teaching. Accordingly, this invention is intended to embrace all alternatives, modifications, and variations that fall within the spirit and broad scope of the [amended] appended claims.

## IN THE ABSTRACT

A directory shell references one or more directories. Each directory comprises a plurality of directory objects each having object attributes and attribute data. One or more categories in the directory shell [is] are associated with one or more directory objects and at least a portion of the associated object attribute. A user interface is operable for a user to query the categories in the directory shell. A directory interface is operable to send a search request to the directories in accordance with the query and receive data from the directories that satisfy the search request.